Minimum Cost Arborescences

Bhaskar Dutta and Debasis Mishra

No 889

WARWICK ECONOMIC RESEARCH PAPERS

DEPARTMENT OF ECONOMICS

THE UNIVERSITY OF WARWICK

Minimum Cost Arborescences*

Bhaskar Dutta[†] Debasis Mishra[‡]

January 8, 2009

Abstract

In this paper, we analyze the cost allocation problem when a group of agents or nodes have to be connected to a *source*, and where the cost matrix describing the cost of connecting each pair of agents is not necessarily symmetric, thus extending the well-studied problem of minimum cost spanning tree games, where the costs are assumed to be symmetric. The focus is on rules which satisfy axioms representing incentive and fairness properties. We show that while some results are similar, there are also significant differences between the frameworks corresponding to symmetric and asymmetric cost matrices.

JEL Classification Numbers: D85, C70

Keywords: directed networks, cost allocation, core stability, continuity, cost monotonicity.

[*] Stimulating conversations about cost allocation problems that one of us have had with Herve Moulin are gratefully acknowledged.

[†] Corresponding author. Department of Economics, University of Warwick, Coventry CV4 7AL, England. Email: `B.Dutta@warwick.ac.uk`

[‡] Indian Statistical Institute, 7 S.J.S. Sansanwal Marg, New Delhi 110016, India. Email: `dmishra@isid.ac.in`

1 Introduction

In a variety of contexts, a group of users may be jointly responsible for sharing the total cost of a joint "project". Often, there is no appropriate "market mechanism" which can allocate the total cost to the individual agents. This has given rise to a large literature which describes axiomatic methods in distributional problems involving the sharing of costs and benefits, the axioms typically representing notions of fairness. In the vast bulk of this literature, the agents have no particular "positional" structure. However, there is a large number of practical problems in which it makes sense to identify the agents with nodes in a graph. Consider, for instance, the following examples.

(i) Multicast routing creates a directed network connecting the source to all receivers; when a packet reaches a branch point in the tree, the router duplicates the packet and then sends a copy over each downstream link. Bandwidth used by a multicast transmission is not directly attributable to any one receiver, and so there has to be a cost-sharing mechanism to allocate costs to different receivers.[1]

(ii) Several villages are part of an irrigation system which draws water from a dam, and have to share the cost of the distribution network. The problem is to compute the minimum cost network connecting all the villages either directly or indirectly to the *source*, i.e. the dam (which is a computational problem), and to distribute the cost of this network amongst the villages.

(iii) In a *capacity synthesis* problem, the agents may share a network for bilateral exchange of information, or for transportation of goods between nodes. Traffic between any two agents i and j requires a certain capacity t_{ij} (width of road, bandwidth). The cost allocation problem is to share the minimum cost of a network in which each pair i and j is connected by a path in which each edge has a capacity of at least t_{ij}. [2]

The combinatorial structure of these problems raises a different set of issues (for instance computational complexity) and proof techniques from those which arise when a network structure is absent. Several recent papers have focused on cost allocation rules appropriate for *minimum cost spanning networks*.[3] In these networks, the agents are each identified with distinct nodes, and there is an additional node (the "source"). Each agent has to be connected either directly or indirectly to the source through some path. A *symmetric* cost

[1] See, for instance, Herzog et al. (1997).

[2] See A. Bogomolnaia et al. (2008), who show that under some assumptions, the capacity synthesis problem is similar, though not identical, to the minimum cost spanning tree problem.

[3] See, for instance, Bird (1976), Bergantinos and Vidal-Puga (2007a), Bergantinos and Vidal-Puga (2007c), Bergantinos and Vidal-Puga (2007b), Bergantinos and Kar (2008), Bogomolnaia and Moulin (2008), Branzei et al. (2004), Feltkamp et al. (1994), Kar (2002), Branzei et al. (2005).

matrix specifies the cost of connecting each pair of nodes. Obviously, the cheapest graph connecting all nodes to the source must be a tree rooted at the node. The cost allocation problem is to assign the total cost of the minimum cost spanning tree to the agents.

In this literature, it is assumed that the cost matrix is symmetric. This implies that the spanning network can be represented as an undirected graph. However, in several situations the cost of connecting agent i to agent j may not be the same as the cost of connecting agent j to agent i. The most obvious examples of this arises in contexts where the geographical position of the nodes affect the cost of connection. For instance, the villages in the second example may be situated at different *altitudes*. In the capacity synthesis problem, the nodes may be towns located along a river, so that transportation costs depend on whether the towns are upstream or downstream. In this paper, we extend the previous analysis by permitting the cost matrix to be asymmetric. The spanning tree will then be a directed graph and is called an *arborescence*. The minimum cost arborescence can be computed by means of an algorithm due to Chu and Liu (1965) and Edmonds (1967). This algorithm is significantly different from the algorithms (Prim's and Kruskal's algorithms) used to compute a minimum cost spanning tree in the symmetric case. However, from a computational perspective, the algorithm for finding a minimum cost arborescence is still a *polynomial time* algorithm.

Our interest is in the cost sharing problem. Following the previous literature, we too focus on an axiomatic approach, the axioms representing a combination of incentive and fairness properties. The first property is the well-known *Stand-Alone Core property*. This requires that no group of individuals be assigned costs which add up to more than the total cost that the group would incur if it built its own subnetwork to connect all members of the group to the source. We provide a constructive proof that the *core* is non-empty by showing that the directed version of the *Bird Rule*,[4] due not surprisingly to the seminal paper of Bird (1976), yields an allocation which belongs to the core of the cost game. Of course, Bird himself had proved the same result when the cost matrix is symmetric. We then prove a partial characterization result which shows that the set of cost allocation rules that are core selections and which satisfy an invariance condition (requiring that the allocation be invariant to costs of edges not figuring in any minimum cost arborescence) assign each individual a cost which lies between the maximum and minimum cost assigned by the set of Bird Rules. This also means that there can be only one such rule when the cost matrix is such that it gives a unique minimum cost arborescence - namely the Bird Rule itself. Of course, this result has no parallel when the cost matrix is symmetric, and emphasizes the difference in the two frameworks.

We then go on to impose two other "minimal" or "basic" requirements - *Continuity*, which requires that the cost shares depend continuously on the cost matrix, and a *Monotonicity*

[4] The Bird Rule is defined with respect to a specific tree, and stipulates that each agent pays the cost of connecting to her predecessor.

requirement which requires that each individual is "primarily" responsible for the cost of his or her incoming edges. We interpret this to mean the following - if the only difference between two cost matrices is that the cost of an incoming edge of agent i goes up, then the cost share of i should (weakly) go up by at least as much as that of any other agent. This property is slightly stronger than the monotonicity condition which was initially defined by Dutta and Kar (2004), and subsequently used in a number of papers on symmetric cost matrices.[5] We construct a rule which satisfies these three basic properties, using Bird's concept of *irreducible* cost matrices. In particular, we show that the Shapley value of the cost game corresponding to the irreducible cost matrix satisfies these three properties. Readers familiar with the literature on the original minimum cost spanning tree problem will immediately recognize that this is exactly the procedure adopted by Bergantinos and Vidal-Puga (2007a) to construct the "folk solution" for minimum cost spanning tree problems.[6] While the folk solution also satisfies the three basic properties, it is important to realize that it belongs to a very different class of rules from the one that we construct in this paper. In particular, Bird's irreducible cost matrix only uses information about the costs of edges figuring in some minimum cost spanning tree - costs of edges not figuring in a minimum cost spanning tree are irrelevant in the construction of the irreducible cost matrix. This obviously means that the folk solution too does not utilize all the information contained in the cost matrix. Indeed, this forms the basis of the critique of "reductionist" solutions (solutions which only utilize information about the costs of edges figuring in some minimum cost spanning tree) by Bogomolnaia and Moulin (2008).

In contrast, the irreducible cost matrix constructed by us (and hence our solution) requires more information than is contained in the minimum cost arborescence(s). This is one important sense in which our solution is qualitatively different from the folk solution. We go on to highlight another important difference. We show that our solution satisfies the directed version of *Ranking*, a property due to Bogomolnaia and Moulin (2008). Our version of Ranking is the following. If the costs of all incoming edges of i are higher than the costs of corresponding edges of j, and the corresponding outgoing edges of i and j are the same, then i should pay strictly more than j. Bogomolnaia and Moulin (2008) point out that all reductionist solutions in the symmetric case - and hence the folk solution - must violate Ranking.

Thus, our results demonstrate that there are significant differences between the frameworks corresponding to symmetric and asymmetric cost matrices, and emphasizes the need

[5] See, for instance, Bergantinos and Vidal-Puga (2007a), Bergantinos and Vidal-Puga (2007c), Bergantinos and Vidal-Puga (2007b).

[6] This is a term coined by Bogomolnaia and Moulin (2008) because this allocation rule has been independently proposed and analyzed in a number of papers. See, for instance, Bergantinos and Vidal-Puga (2007a), Bergantinos and Vidal-Puga (2007c), Bergantinos and Vidal-Puga (2007b), Bogomolnaia and Moulin (2008), Branzei et al. (2004), Feltkamp et al. (1994), Norde et al. (2001) Branzei et al. (2005).

for more systematic analysis of the cost allocation problem for minimum cost arborescences.

2 Framework

Let $N = \{1, 2, \ldots, n\}$ be a set of n agents. We are interested in *directed graphs* or *digraphs* where the nodes are elements of the set $N^+ \equiv N \cup \{0\}$, where 0 is a distinguished node which we will refer to as the *source*. We assume that the set of edges of this graph is the set $\{ij : i \in N^+, j \in N\}$. So, we assume that there is an edge from every $i \in N^+$ to every $j \in N$. We will also have to consider digraphs on some subsets of N^+. So, for any set $S \subset N$, let S^+ denote the set $S \cup \{0\}$. Then, a digraph on S^+ consists of a set of *directed edges* out of the set $\{ij : i \in S^+, j \in S\}$.

A typical graph[7] over S^+ will be represented by g_S whose edges are out of the set $\{ij : i \in S^+, j \in S\}$. When there is no ambiguity about the set S (usually when we refer to a graph on N^+), we will simply write g, g' etc instead of g_S, g'_S. We will denote the set of *incoming edges* of a node i in graph g as $\delta^-(i, g) = \{ji : ji \text{ is an edge in } g\}$. Similarly, the set of *outgoing edges* of a node i is denoted as $\delta^+(i, g) = \{ij : ij \text{ is an edge in } g\}$.

A *cost matrix* $C = (c_{ij})$ for N^+ represents the cost of various edges which can be constructed from nodes in N^+. That is, c_{ij} is the cost of the edge ij. We assume that each $c_{ij} \geq 0$ for all ij. We also adopt the convention that for each $i \in N^+$, $c_{ii} = 0$. Note that the cost of an edge ij need not be the same as that of the edge ji - the direction of the edge does matter.[8] Given our assumptions, each cost matrix is nonnegative, and of order $n + 1$. The set of all cost matrices for N is denoted by $\mathcal{C}_N$. For any cost matrix C, denote the cost of a graph g as $c(g)$. That is,

$$c(g) = \sum_{ij \in g} c_{ij}$$

A *path* in g is a sequence of distinct nodes $(i^1, \ldots, i^K)$ such that $i^j i^{j+1}$ is an edge in g for all $1 \leq j \leq K - 1$. If $(i^1, \ldots, i^K)$ is a path, then we say that it is a path from i^1 to i^K using edges $i^1 i^2, i^2 i^3, \ldots, i^{K-1} i^K$. A *cycle* in g is a sequence of nodes $(i^1, \ldots, i^K, i^{K+1})$ such that $(i^1, \ldots, i^K)$ is a path in g, $i^K i^{K+1}$ is an edge in g, and $i^1 = i^{K+1}$.

A node i is *connected* to node j if there is a path from node j to node i. Our interest is in graphs in which every agent in N is connected to the source 0.

Definition 1 *A graph g is an arborescence rooted at* 0 *for N if and only if g contains no cycle and $\delta^-(i, g) = 1$ for all $i \in N$.*

[7] Henceforth, we will use the term “graph” to denote digraphs. Similarly, we will use the term “edge” to denote a directed edge.

[8] As the reader will recognize, this distinguishes our approach from the literature on minimum cost spanning tree problems.

Let A_N be the set of all arborescences for N. The *minimum cost arborescence* (MCA) corresponding to cost matrix C is an arborescence g such that $c(g) \leq c(g')$ for all $g' \in A_N$.

Let $M(C)$ denote the set of minimum cost arborescences corresponding to the cost matrix C for the set N, and $T(C)$ represent the total cost associated with any element g of $M(C)$. While our main interest is in minimum cost arborescences for N, we will also need to define the minimum cost of connecting *subsets* of N to the source 0. The set of arborescences for any subset S of N will be denoted A_S and the set of minimum cost arborescences will be represented by $M(C, S)$. Also, $T_S(C)$ will denote the total cost associated with any element of $M(C, S)$.

Clearly, a minimum cost arborescence is analogous to a minimum cost spanning tree (MCST) for undirected graphs. Alternatively, an MCA may be viewed as a generalization of an MCST when the cost matrix is not symmetric.

2.1 The Recursive Algorithm

It turns out that the typical greedy algorithms used to construct minimum cost spanning trees fail to generate minimum cost arborescences. Figure 1 illustrates this phenomenon.[9] Recall that a unique feature of minimum cost spanning trees is that an MCST must always choose the minimum cost (undirected) edge corresponding to any cost matrix. Notice, however, that in Figure 1, the minimum cost arborescence involves edges $01, 12, 23$. But it does not involve the minimum cost edge 31.

We now describe an algorithm due to Chu and Liu (1965) and Edmonds (1967) to construct an MCA. Though the algorithm is quite different from the algorithms for constructing an MCST, it is still computationally tractable as it runs in *polynomial time*.

The algorithm works *recursively*. In each recursion stage, the original cost matrix on the original set of nodes and the original graph is transformed to a new cost matrix on a new set of nodes and a new graph. The terminal stage of the recursion yields an MCA for the terminal cost matrix and the terminal set of nodes. One can then "go back" through the recursion stages to get an MCA for the original problem. Since the algorithm to compute an MCA is different from the algorithm to compute an MCST, we first describe it in detail with an example.

Consider the example in Figure 2. To compute the MCA corresponding to the cost matrix in Figure 2, we first perform the following operation: for every node, subtract the value of the minimum cost incident edge from the cost of every edge incident on that node. As an example, 31 is the minimum cost incident edge on node 1 with cost 1. Hence, the new cost of edge 31 is $1 - 1 = 0$, edge 01 is $2 - 1 = 1$, and edge 21 is $3 - 1 = 2$. In a similar fashion,

[9] The numbers besides each edge represent the cost of the edge. The missing edges have very high cost and do not figure in any MCA.

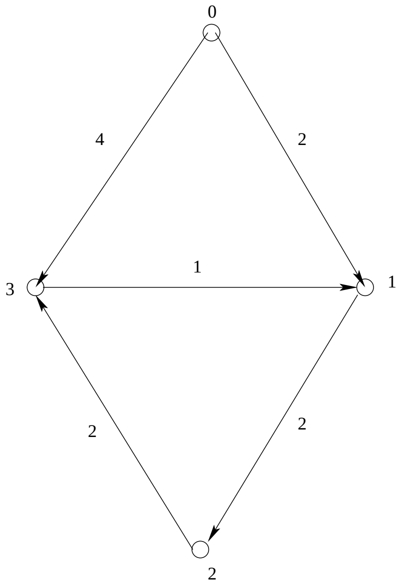

Figure 1: An example where greedy algorithm of undirected graph fails

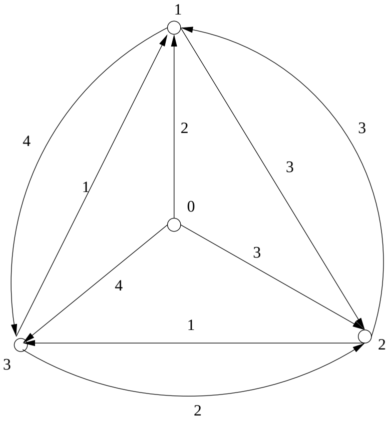

Figure 2: An example

compute the following new cost matrix C^1:

$$c^1_{01} = 1,\ c^1_{21} = 2,\ c^1_{31} = 0$$
$$c^1_{02} = 1,\ c^1_{12} = 1,\ c^1_{32} = 0$$
$$c^1_{03} = 3,\ c^1_{23} = 0,\ c^1_{13} = 3.$$

Clearly, an MCA corresponding to C^1 is also an MCA corresponding to the original cost matrix. So, we find an MCA corresponding to C^1. To do so, for every node, we pick a zero cost edge incident on it (by the construction of C^1, there is at least one such edge for every node). If such a set of edges form an arborescence, it is obviously an MCA corresponding to C^1, and hence, corresponding to the original cost matrix. Otherwise, cycles are formed by such a set of zero cost edges. In the example, we see that the set of minimum cost edges are 31, 32, and 23. So, 32 and 23 form a cycle. The algorithm then merges nodes 2 and 3 to a single *supernode* (23), and constructs a new graph on the set of nodes 0, 1, and supernode (23). We associate a new cost matrix $\widetilde{C}^1$ on this set of nodes using C^1 as follows: $\tilde{c}^1_{01} = c^1_{01} = 1$; $\tilde{c}^1_{0(23)} = \min\{c^1_{02}, c^1_{03}\} = \min\{1,3\} = 1$; $\tilde{c}^1_{1(23)} = \min\{c^1_{12}, c^1_{13}\} = \min\{1,3\} = 1$, (23)1, $\tilde{c}^1_{(23)1} = \min\{c^1_{21}, c^1_{31}\} = \min\{2,0\} = 0$. The graph consisting of these edges along with the costs corresponding to cost matrix $\widetilde{C}^1$ is shown in Figure 3.

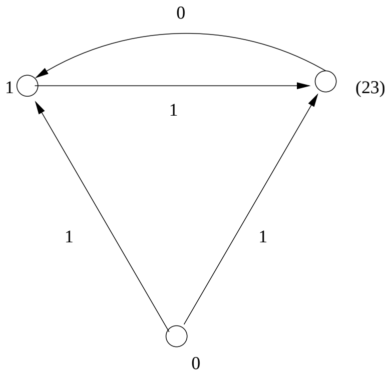

Figure 3: After first stage of the algorithm

We now seek an MCA for the situation depicted in Figure 3. We repeat the previous step. The minimum cost incident edge on 1 is (23)1 and we choose the minimum cost incident edge on (23) to be 0(23). Subtracting the minimum costs as we did earlier, we get that 0(23) and (23)1 are edges with zero cost. Since these edges form an arborescence, this is an MCA corresponding to cost matrix $\widetilde{C}^1$. To get the MCA for the original cost matrix, we note that $\tilde{c}^1_{0(23)} = c^1_{02}$. Hence, 0(23) is replaced by edge 02. Similarly, (23)1 is replaced by edge 31. The cycle in supernode (23) is broken such that we get an arborescence - this can be done by choosing edge 23 since 02 is the incident edge on supernode (23). Hence, the MCA corresponding to the original cost matrix is: 02, 23, 31.

We now describe the algorithm formally. Given any cost matrix C, we will say that a set of nodes $I = \{1, \ldots, K\}$ form a C-*cycle* if $c_{i,i+1} = 0$ for all $i = 1, \ldots K-1$ and $c_{K1} = 0$.

- *Stage* 0: Set $C^0 \equiv \widetilde{C}^0 \equiv C$, $N^0 \equiv N \cup \{0\}$, and for each $j \in N$,

$$\Delta^0_j = \min_{i \neq j} c^0_{ij}, N^0_j = \{j\}.$$

- *Stage* 1: For each pair i, j, define $c^1_{ij} = c^0_{ij} - \Delta^0_j$. Construct a partition $\{N^1_1, \ldots, N^1_{K^1}\}$ of N^0 such that each N^1_k is either a C^1-cycle of elements of N^0 or a singleton with the restriction that no set of singletons forms a C^1-cycle.[10] Denote $N^1 = \{N_1^1, \ldots, N_{K^1}^1\}$. For each $k, l \in \{1, \ldots, K^1\}$, define

$$\tilde{c}^1_{N_k^1 N_l^1} = \min_{i \in N_k^1, j \in N_l^1} c^1_{ij} = \min_{i \in N_k^1, j \in N_l^1} [\tilde{c}^0_{ij} - \Delta^0_j].$$

Hence, $\widetilde{C}^1$ is a cost matrix on nodes N^1. For each $k \in \{1, \ldots, K^1\}$, define

$$\Delta^1_k = \min_{i \notin N_k^1, j \in N_k^1} c^1_{ij} = \min_{N_l^1 \neq N_k^1} \tilde{c}^1_{N_l^1 N_k^1}.$$

- *Stage t*: For each pair i, j, define $c^t_{ij} = c^{t-1}_{ij} - \Delta^{t-1}_j$. Construct a partition $\{N_1^t, \ldots, N_{K^t}^t\}$ of N^{t-1} such that each N_k^t is either a C^t-cycle of elements of N^{t-1} or a singleton element of N^{t-1} with the restriction that no set of singletons forms a C^t-cycle. For each $k, l \in \{1, \ldots, K^t\}$, define

  $$\tilde{c}^t_{N_k^t, N_l^t} = \min_{i \in N_k^t, j \in N_l^t} c^t_{ij} = \min_{N_p^{t-1} \subseteq N_k^t, N_q^{t-1} \subseteq N_l^t} [\tilde{c}^{t-1}_{N_p^{t-1} N_q^{t-1}} - \Delta^{t-1}_q].$$

  Note that $\widetilde{C}^t$ is a cost matrix on nodes N^t.

  For each $k \in \{1, \ldots, K^t\}$, define

  $$\Delta^t_k = \min_{i \notin N_k^t, j \in N_k^t} c^t_{ij} = \min_{N_l^t \neq N_k^t} \tilde{c}^t_{N_l^t N_k^t}.$$

Terminate the algorithm at stage T if each N_k^T is a singleton. That is, each N_k^T coincides with some N_k^{T-1}, so that each N_k^T is also an element of the partition N^{T-1}. Since the source cannot be part of any cycle and since N is finite, the process must terminate - there must indeed be a stage when each N_k^T is a singleton.

We will sometimes refer to sets of nodes such as N_k^t as *supernodes.* The algorithm proceeds as follows. First, construct a MCA g^T of graph with nodes N^T (or equivalently N^{T-1}) corresponding to cost matrix $\widetilde{C}^T$. By definition, this can be done easily by choosing a minimum cost edge for every node in N^T. Then, unless $T = 1$, "extend" g^t to g^{t-1} for all $2 \leq t \leq T$ by establishing connections between the elements of N^t and N^{t-1}, until we reach $g^1 \in M(C)$.

Since g^T is an MCA on N^T corresponding to cost matrix $\widetilde{C}^T$, if $T = 1$, then g^1 is the MCA on N^0 corresponding to cost matrix C^1, and hence C.

Suppose $T \geq 2$. For every $2 \leq t \leq T$, given an MCA g^t corresponding to $\widetilde{C}^t$, we extend it to an MCA g^{t-1} on nodes N^{t-1} corresponding to $\widetilde{C}^{t-1}$ as follows.

(i) For every edge $N_i^t N_j^t$ in g^t, let $\tilde{c}^t_{N_i^t, N_j^t} = \tilde{c}^{t-1}_{N_p^{t-1} N_q^{t-1}} - \Delta^{t-1}_q$ for some $N_p^{t-1} \subseteq N_i^t$ and some $N_q^{t-1} \subseteq N_j^t$. Then, replace $N_i^t N_j^t$ with $N_p^{t-1} N_q^{t-1}$ in graph g^{t-1}.

[10] If there is some node $j \in N$ such that two or more edges minimize cost, then break ties arbitrarily.

(ii) In the previous step, if we choose $N_p^{t-1}N_q^{t-1}$ as an incoming edge to N_k^t, which is a C^{t-1} cycle, then retain all the edges in g^{t-1} which constitute the cycle in supernode N_k^t except the edge which is incoming to N_q^{t-1}.

This completes the extension of g^{t-1} from g^t. It is not difficult to see that g^{t-1} is a MCA for graph with nodes N^{t-1} corresponding to $\widetilde{C}^{t-1}$ (for a formal argument, see Edmonds (1967)). Proceed in this way to $g^1 \in M(C)$.

2.2 The Cost Allocation problem

Given any cost matrix C, the recursive algorithm will generate an MCA corresponding to C, and hence the total cost of connecting all the agents in N to the source. Since the total cost is typically less than the direct cost of connecting each agent to the source, the group as a whole gains from cooperation. So, there is the issue of how to distribute the cost savings amongst the agents or, what is the same thing, how to allocate the total cost to the different agents.

Definition **2** *A cost allocation rule is a function* $\mu : \mathcal{C}_N \rightarrow \Re^N$ *satisfying* $\sum_{i \in N} \mu_i(C) = T(C) \;\; \forall \; C \in \mathcal{C}_N$.

So, for each cost matrix, a cost allocation rule specifies how the total cost of connecting all agents to the source should be distributed. Notice that our definition incorporates the notion that the rule should be *efficient* - the costs distributed should be exactly equal to the total cost.

In this paper, we follow an axiomatic approach in defining "fair" or "reasonable" cost allocation rules. The axioms that we will use here reflect a concern for both "stability" and fairness.

The notion of stability reflects the view that any specification of costs must be acceptable to all groups of agents. That is, no coalition of agents should have a justification for feeling that they have been overcharged. This leads to the notion of the *core* of a specific cost allocation game.

Consider any cost matrix C on N^+. While the set of agents incur a total cost of $T(C)$ to connect each node to the source, each subset S of N incurs a corresponding cost of $T_S(C)$. It is natural to assume that agents in any subset S will refuse to cooperate if an MCA for N is built and they are assigned a total cost which exceeds $T_S(C)$ - they can issue the credible threat of building their own MCA.

So, each cost matrix C yields a cost game (N, c) where

$$\text{For each } S \subseteq N, c(S) = T_S(C)$$

The *core* of a cost game (N, c) is the set of all allocations x such that

$$\text{for all } S \subseteq N, \sum_{i \in S} x_i \leq c(S), \sum_{i \in N} x_i = c(N)$$

We will use $Co(N, C)$ to denote the core of the cost game corresponding to C.

Definition **3** *A cost allocation rule μ is a Core Selection (CS) if for all C, $\mu(C) \in Co(N, C)$.*

A rule which is a core selection satisfies the intuitive notion of stability since no group of agents can be better off by rejecting the prescribed allocation of costs.

The next axiom is essentially a property which helps to minimize the computational complexity involved in deriving a cost allocation. The property requires the cost allocation to depend only on the costs of edges involved in the MCA s. That is, if two cost matrices have the same set of MCA s, and the costs of edges involved in these trees do not change, then the allocation prescribed by the rule should be the same.

For any N, say that two cost matrices C, C' are *arborescence equivalent* if (i) $M(C) = M(C')$, and (ii) if ij is an edge in some MCA, then $c_{ij} = c'_{ij}$.

Definition **4** *A cost allocation rule μ satisfies Independence of Irrelevant Costs (IIC) if for all C, C', $\mu(C) = \mu(C')$ whenever C and C' are arborescence-equivalent.*

The next axiom is straightforward.

Definition **5** *A cost allocation rule μ satisfies Continuity (Con) if for all N, μ is continuous in the cost matrix C.*

In the present context, a fundamental principle of fairness requires that each agent's share of the total cost should be monotonically related to the vector of costs of its own incoming edges. So, if the cost of say edge ij goes up, and all other edges cost the same, then j's share of the total cost should not go down. This requirement is formalised below.

Definition **6** *A cost allocation rule satisfies Direct Cost Monotonicity (DCM) if for all C, C', if $c_{ij} < c'_{ij}$, and for all other edges $kl \neq ij$, $c_{kl} = c'_{kl}$, then $\mu_j(C') \geq \mu_j(C)$.*

This is the counterpart of the assumption of Cost Monotonicity introduced by Dutta and Kar (2004). Clearly, DCM is also compelling from the point of view of incentive compatibility. If DCM is not satisfied, then an agent has an incentive to inflate costs (assuming an agent is responsible for its incoming edge costs).

Notice that DCM permits the following phenomenon. If the cost of some edge ij goes up (other edge costs remain the same), then the cost allocated to j may go up by say ϵ, but the cost allocated to some k may go up by more than ϵ. This is clearly unfair. The next axiom rules out this possibility.

Definition 7 *A cost allocation rule satisfies Direct Strong Cost Monotonicity (DSCM) if for all N, and for all C, C', if $c_{ij} < c'_{ij}$ and $c_{kl} = c'_{kl}$ for all $kl \neq ij$ we have $\mu_j(C) - \mu_j(C') \geq \mu_k(C) - \mu_k(C')$ for all k.*

Note that DSCM implies DCM.

Our next axiom is a symmetry condition which essentially requires that if two nodes i and j have identical vectors of costs of incoming and outgoing edges, then their cost shares should not differ.

Definition 8 *A cost allocation rule μ satisfies Equal Treatment of Equals (ETE) if for all $i, j \in N$ and for all cost matrices C with $c_{ki} = c_{kj}$ and $c_{ik} = c_{jk}$ for all $k \neq i, j$ and $c_{ij} = c_{ji}$, we have $\mu_i(C) = \mu_j(C)$.*

Our final axiom is that of *Ranking*, adapted from Bogomolnaia and Moulin (2008). Ranking compares cost shares across individual nodes and insists that if costs of all incoming edges of i are uniformly higher than the corresponding costs for j, while the costs of outgoing edges are the same, then i should pay strictly more than j. Notice that it is similar in spirit to DSCM in that both insist that nodes are "primarily" responsible for their incoming costs.

Definition 9 *A cost allocation rule μ satisfies Ranking (R) if for all $i, j \in N$ and for all cost matrices C with $c_{ik} = c_{jk}$ and $c_{ki} > c_{kj}$ for all $k \neq i, j$ and $c_{ji} > c_{ij}$, we have $\mu_i(C) > \mu_j(C)$.*

3 A Partial Characterization Theorem

In the context of minimum cost spanning tree problems, Bird (1976) is a seminal paper. Bird defined a specific cost allocation rule - the Bird Rule, and showed that the cost allocation specified by his rule belonged to the core of the cost game, thereby providing a constructive proof that the core is always non-empty.[11]

In this section, we show that even in the directed graph context, the Bird allocations belong to the core of the corresponding game. We then show that if a cost allocation rule satisfies IIC and CS, then the cost allocation of each agent must lie between the minimum and maximum costs paid by the individual in different Bird allocations. In particular, this implies that the cost allocation rule must coincide with the Bird Rule on the set of cost matrices which give rise to unique MCA s.

For any arborescence $g \in A_N$, for any $i \in N$, let $\rho(i)$ denote the predecessor of i in g. That is, $\rho(i)$ is the unique node which comes just before i in the path connecting i to the source 0.

[11] Bird defined his rule for a specific MCST. Since a cost matrix may have more than one MCST, a proper specification of a "rule" can be obtained by, for instance, taking a convex combination of the Bird allocations obtained from the different minimum cost spanning trees.

Definition **10** *Let C be some cost matrix.*

(i) A Bird allocation of any MCA $g(C)$ is $b_i(g,C) = c_{\rho(i)i}$.

(ii) A Bird rule is given by $B_i(C) = \sum_{g \in M(C)} w_g b_i(g,C)$, where $\sum_{g \in M(C)} w_g = 1$ and $w_g \geq 0$ for each $g \in M(C)$.

Notice that the Bird rule is a family of rules since it is possible to have different convex combinations of Bird allocations. We first prove that Bird allocations belong to the core of the cost game.

Theorem **1** *For every cost matrix C and MCA $g(C)$, $b(g,C) \in Co(N,C)$.*

Proof: Let g^* be the outcome of the recursive algorithm supplemented by tie-breaking rules if necessary. Let x denote the corresponding Bird allocation.

Let $\{N_1^1, \ldots, N_L^1\}$ be the set of supernodes after stage 1 of the recursive algorithm. If the algorithm terminates in one step, then each N_k^1 contains only a singleton node. Otherwise, at least one N_k^1 contains more than one node of N. For each N_k^1, let i_k denote the node at which g^* enters N_k^1. Then, the following is true for every N_k^1 and for every $i \neq i_k$,

$$x_i = \Delta_i^0 \text{ and } x_{i_k} \geq \Delta_{i_k}^0 \tag{1}$$

Now, let S be a *maximal* blocking coalition, so that $c(S) < \sum_{i \in S} x_i$.

Let $K = \{k | S \cap N_k^1 \neq \emptyset\}$. We first show that if $k \in K$, then $N_k^1 \subset S$. For suppose $T = N_k^1 \setminus S$, and T is nonempty. Now,

$$\begin{aligned} c(S \cup T) &= c(S) + \sum_{i \in T} \Delta_i^0 \\ &\leq c(S) + \sum_{i \in T} x_i \\ &< \sum_{i \in S \cup T} x_i \end{aligned}$$

where the first inequality follows from (1) and the last inequality from the fact that S is a blocking coalition. This shows that $S \cup T$ is a blocking coalition, violating maximality of S.

Hence,

$$\cup_{k \in K} N_k^1 = S$$

Since N is not a blocking coalition, there must be some $N_j^1 \neq 0$ such that $j \notin K$. Call N_i^1 a predecessor of N_j^1 if the incoming edge (of g^*) to N_j^1 is from N_i^1.[12] Moreover, we can

[12] Note that each N_j^1 has a unique predecessor.

choose some $j \notin K$ such that its predecessor is either contained in S or it is the source 0. Then,

$$c(S \cup N_j^1) \leq c(S) + \sum_{i \in N_j^1} x_i$$

But, then $S \cup N_j^1$ is a blocking coalition, violating the maximality of S. This contradiction shows that there cannot be any blocking coalition. ■

Of course, the Bird Rule satisfies IIC. We now prove a partial converse by showing that any cost allocation rule satisfying CS and IIC must specify cost shares which are bounded by the minimum and maximum Bird allocations. That is, for each C and each $i \in N$, let

$$\begin{aligned} b_i^m(C) &= \min_{g \in M(C)} b_i(g, C) \\ b_i^M(C) &= \max_{g \in M(C)} b_i(g, C). \end{aligned}$$

Then, we have the following theorem.

Theorem **2** *If μ is a cost allocation rule satisfying CS and IIC, then for all C, for all $i \in N$, $\mu_i(C) \in [b_i^m(C), b_i^M(C)]$.*

Proof: Fix C and consider any μ which satisfies CS and IIC. Let $\mu(C) \equiv x$.

(i) Suppose there is $i \in N$ such that $x_i = b_i^m(C) - \epsilon$ for some $\epsilon > 0$. Let g be the MCA which gives rise to the Bird allocation in which i pays $b_i^m(C)$, and let $b(g, C)$ be the Bird allocation. Let S be the set of successors of i - these are the nodes j such that i lies in the path from 0 to j.

Now, let $T = N \setminus (S \cup \{i\})$. We first show that $S \neq \emptyset$. Assume for contradiction $S = \emptyset$. Then, i is the leaf of the MCA g. Hence, if we remove the edge which goes to i, we get an MCA of coalition $N \setminus \{i\}$. This means, $c(N \setminus \{i\}) = c(N) - b_i(g, C) < \sum_{j \in N} x_j - x_i = \sum_{j \in N \setminus \{i\}} x_j$, implying that $N \setminus \{i\}$ will block, which is a contradiction. Now, consider the following.

$$\begin{aligned} \sum_{j \in S} x_j &= \sum_{j \in N} b_j(g, C) - \sum_{j \in T} x_j - x_i \\ &> \sum_{j \in N} b_j(g, C) - \sum_{j \in T} x_j - b_i(g, C) \\ &\geq \sum_{j \in N} b_j(g, C) - c(T) - b_i(g, C) \qquad \text{(By CS)} \\ &\geq \sum_{j \in N} b_j(g, C) - \sum_{j \in T} b_j(g, C) - b_i(g, C) \\ &= \sum_{j \in S} b_j(g, C). \end{aligned}$$

Hence, $\sum_{j\in S} x_j > \sum_{j\in S} b_j(g,C)$. Let $S' = \{j \in S|\rho(j) = i\}$ be the set of immediate successors of i. Let $|S'| = s$. Also, denote $\rho(i) \equiv k$. Then, consider a cost matrix C' such that

$$\begin{aligned}
\text{for all } j \in S', c'_{kj} &= c_{ij} + \frac{\epsilon}{2s} \qquad \forall\, j \in S' && (2)\\
\text{for all other edges } kl, c'_{kl} &= c_{ij}. && (3)
\end{aligned}$$

Note that $M(C) = M(C')$. Also, the edges whose costs are changed do not figure in any MCA. So, from IIC, $\mu(C') = \mu(C) = x$. Now, consider the following arborescence for nodes in $N \setminus \{i\}$: take the MCA g and disconnect node i, i.e., delete all edges outgoing from node i and the unique edge incident on i, and connect nodes in S' using the direct edge from k to every node in S'. This is an arborescence because every node is still connected to the source and there are no cycles (if there was a cycle here, we would get a cycle in g via i). The cost of this arborescence is $c(N) - b_i(g,C) + \frac{\epsilon}{2}$. This gives,

$$\begin{aligned}
c'(N \setminus \{i\}) &\leq c(N) - b_i(g,C) + \frac{\epsilon}{2}\\
&< c(N) - b_i(g,C) + \epsilon\\
&= c(N) - x_i\\
&= \sum_{j\in N\setminus\{i\}} x_j
\end{aligned}$$

This shows that $x \notin Co(N,C')$, contradicting the assumption that μ satisfies CS.

(ii) Suppose μ satisfies CS and IIC, and there is some cost matrix C such that

$$\mu_i(C) = b_i^M(C) + \epsilon \text{ for some } i \in N$$

Then, consider C' such that $c'_{0i} = b_i^M(C) + \epsilon/2$, while the cost of all other edges remains the same. The change in the cost of the edge $0i$ cannot affect the set of MCA s, and so $\mu(C') = \mu(C)$ by IIC. But, $\{i\}$ will block $\mu(C')$ and so μ cannot be in the core $Co(N,C')$. ■

Remark 1 *Note that for any cost matrix C, if $M(C)$ is a singleton, then any cost allocation rule μ satisfying CS and IIC must coincide with the unique Bird Rule.*

The following related theorem is of independent interest. We show that no *reductionist* solution satisfying CS can satisfy either Con or DCM. As we have pointed out in the introduction, this result highlights an important difference between the solution concepts for the classes of symmetric and asymmetric cost matrices.

THEOREM **3** *Suppose a cost allocation rule satisfies CS and IIC. Then, it cannot satisfy either Con or DCM.*

Proof: Let $N = \{1, 2\}$. Consider the cost matrix given below

$$c_{01} = 6, c_{02} = 4, c_{12} = 1, c_{21} = 3$$

Then,

$$b_1^m(C) = 3, b_2^m(C) = 1, b_1^M(C) = 6, b_2^M(C) = 4$$

Let μ satisfy IIC and CS. Then,

$$\mu_1(C) \in [3, 6], \mu_2(C) \in [1, 4]$$

Now, consider C' such that $c'_{01} = 6 + \epsilon$ where $\epsilon > 0$, and all other edges cost the same as in C. There is now a unique MCA, and so since μ satisfies CS and IIC,

$$\mu(C') = (3, 4)$$

If μ also satisfies DCM, then we need $\mu_1(C') \geq \mu_1(C)$. This implies

$$\mu(C) = (3, 4)$$

Now consider C'' such that $c''_{02} = 4 + \gamma$, with $\gamma > 0$ while all other edges cost the same as in C. It follows that if μ is to satisfy CS, IIC, and DCM, then

$$\mu(C) = (6, 1)$$

This contradiction establishes that there is no μ satisfying CS, IIC, and DCM. Now, if μ is to satisfy CS, IIC and Con, then there must be a continuous function $f : \Re^2 \to \Re^2$ such that

$$(3, 4) = \mu(C) + f(\epsilon, 0), \text{ and } (6, 1) = \mu(C) + f(0, \gamma)$$

for all $\epsilon, \gamma > 0$ and with $f(0, 0) = (0, 0)$. Clearly, no such continuous function can exist. ■

The last theorem and the earlier remark demonstrate that there is a sharp difference between allocation rules when the cost matrix is symmetric and when it is asymmetric. The literature on minimum cost spanning tree games shows that there are a large number of cost allocation rules satisfying CS, IIC and monotonicity and\or continuity. Clearly, options are more limited when the cost matrix is asymmetric. Nevertheless, we show in the next section that it is possible to construct a rule satisfying the three "basic" properties of CS, DCM and Con. Of course, the rule we construct will not satisfy IIC.

4 A Rule Satisfying CS, DSCM and Con

In this section, we construct a rule satisfying the three "basic" axioms of CS, DSCM and Con. Our construction will use a method which has been used to construct the "folk solution". The rule satisfies counterparts of the three basic axioms in the context of the minimum cost spanning tree framework. However, the rule constructed by us will be quite different. In particular, the "folk solution" satisfies the counterpart of IIC but does not satisfy Ranking for symmetric cost matrices. In contrast, while our rule obviously cannot satisfy IIC, we will show that it satisfies Ranking.

We first briefly describe the methodology underlying the construction of the "folk solution".

Bird (1976) defined the concept of an *irreducible* cost matrix corresponding to any cost matrix C in the minimum cost spanning tree problem. The irreducible matrix C^R is obtained from a symmetric cost matrix C in the following way. Let g be some minimum cost spanning tree for a symmetric cost matrix C.[13] For any $i, j \in N$, let $P(ij)$ denote the set of paths from i to j for this tree. Then, the "irreducible" cost of the edge ij is

$$c^R_{ij} = \min_{p(ij) \in P(i,j)} \max_{kl \in p(i,j)} c_{kl} \qquad \forall\, i, j \in N.$$

Of course, if some ij is part of a minimum cost spanning tree, then $c^R_{ij} = c_{ij}$. So, while every original minimum cost spanning tree remains a minimum cost spanning tree for the irreducible cost matrix, new trees also minimise the irreducible cost of a spanning tree.

Bird (1976) showed that the cost game corresponding to this C^R is *concave*. From the well-known theorem of Shapley (1971), it follows that the Shapley value belongs to the core of this game. Moreover, since $c^R_{ij} \leq c_{ij}$ for all ij, it follows that the core of the game corresponding to C^R is contained in the core of the game corresponding to C. So, the cost allocation rule choosing the Shapley value of the game corresponding to C^R satisfies CS. Indeed, it also satisfies Cost Monotonicity[14] and Continuity.

It is natural to try out the same approach for the minimum cost arborescence problem. However, a similar approach cannot work when the cost matrix is asymmetric. Notice that the construction of the irreducible cost matrix outlined above only uses information about the costs of edges belonging to some minimum cost spanning tree. So, the Shapley value of the cost game corresponding to the irreducible cost matrix must also depend only on such information. In other words, the "folk" solution must satisfy IIC. It follows from Theorem 3, that no close cousin of the folk solution can satisfy the desired properties.

[13] If more than one tree minimises total cost, it does not matter which tree is chosen.

[14] Bergantinos and Vidal-Puga (2007a) show that it satisfies a stronger version of cost monotonicity - a *solidarity* condition which requires that if the cost of some edge goes up, then the cost shares of all agents should (weakly) go up.

In what follows, we focus on the essential property of an irreducible cost matrix - it is a cost matrix which has the property that the cost of no edge can be reduced any further if an MCA for the original matrix is to remain an MCA for the irreducible matrix and the total cost of an MCA of the irreducible cost matrix is the same as that of the original matrix. We use the recursive algorithm to construct an irreducible matrix.

Given any cost matrix C, we construct an irreducible cost matrix C^R, and show that C^R possesses the following properties.

1. The irreducible cost matrix C^R is well-defined in the sense that it does not depend on the tie-breaking rule used in the recursive algorithm.
2. *Every* edge ij is part of some MCA corresponding to C^R and $T(C) = T(C^R)$.
3. The cost game corresponding to C^R is concave.

We will use these properties to show that the rule choosing the Shapley value of the cost game corresponding to C^R satisfies CS, DSCM, and Con.

We first illustrate the construction of the irreducible cost matrix for the example in Figure 2. Like in the recursive algorithm, we do it recursively. So, we first construct an irreducible cost matrix on nodes N^T and corresponding cost matrix $\widetilde{C}^T$ of the last stage T of the algorithm. For the example in Figure 2, Figure 3 exhibits the graph on this set of nodes. An irreducible cost matrix corresponding to this cost matrix can be obtained as follows: set the cost of any edge ij equal to the minimum cost incident edge on j. Denoting this reduced cost matrix as $\widetilde{C}^R$, we get $\tilde{c}^R_{01} = \tilde{c}^R_{(23)1} = 0$ and $\tilde{c}^R_{0(23)} = \tilde{c}^R_{1(23)} = 1$. It is easy to verify that this is indeed an irreducible cost matrix. Now, we extend $\widetilde{C}^R$ to an irreducible cost matrix C^R corresponding to the original cost matrix. For any edge ij, if i and j belong to different supernodes N^1_i and N^1_j respectively in Figure 3, then $c^R_{ij} = \tilde{c}^R_{N^1_i, N^2_j} + \Delta^0_j$. For example, $c^R_{12} = \tilde{c}^R_{1(23)} + 2 = 1 + 2 = 3$ and $c^R_{03} = \tilde{c}^R_{0(23)} + 1 = 1 + 1 = 2$. For any ij, if i and j belong to the same supernode, then $c^R_{ij} = \Delta^0_j$. For example, $c^R_{23} = 1$ and $c^R_{32} = 2$. Using this, we show the irreducible cost matrix in Figure 4.

We now formalize these ideas of constructing an irreducible cost matrix below.

Fix some cost matrix C. Suppose that the recursive algorithm terminates in T steps given some tie-breaking rule. Say that $i, j \in N$ are t-neighbours if $i, j \in N^t_k \in N^t$ for some $t \leq T - 1$ and there is no $t' < t$, $N^{t'}_k$ such that $i, j \in N^{t'}_k$. If i, j are not t-neigbours for any $t \leq T - 1$, then call them T-neighbours.

If $i \in N^t_k$, then let

$$\delta^t_i = \Delta^t_k.$$

For any pair $i, j \in N$ which are t-neighbours,

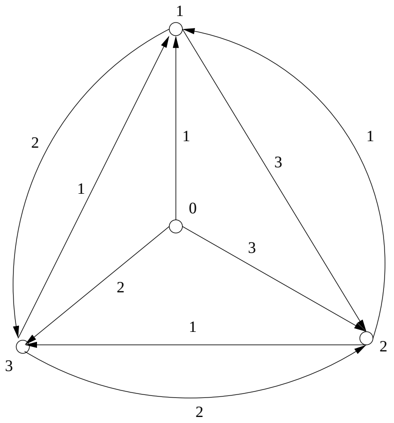

Figure 4: Irreducible cost matrix for graph in Figure 2

$$c_{ij}^R = \sum_{t'=0}^{t-1} \delta_j^{t'} \tag{4}$$

So, for instance, if the algorithm terminates in one step, then all i, j are 1-neighbours, and

$$c_{ij}^R = \min_{k \neq j} c_{kj} \equiv \Delta_j^0$$

Since the recursive algorithm breaks ties arbitrarily and the irreducible cost matrix uses the recursive algorithm, it does not follow straightaway that two different tie-breaking rules result in the same irreducible cost matrix. We say that the irreducible cost matrix is *well-defined* if different tie-breaking rules yield the same irreducible cost matrix.

LEMMA **1** *The irreducible cost matrix defined through Equation (4) is well-defined.*

Proof: To simplify notation, suppose there is a tie in step one of the algorithm, between exactly two edges say ij and kj, as the minimum incident cost edge on node j.[15] Hence, $c_{ij} = c_{kj} = \Delta_j^0$. We will investigate the consequence of breaking this tie one way or the other. For this, we break the other ties in the algorithm exactly the same way in both cases.

We distinguish between three possible cases.

CASE 1: Suppose j forms a singleton node N_j^1 in step 1 irrespective of whether the algorithm breaks ties in favour of ij or kj. Then, in either case $\Delta_j^1 = 0$. Moreover, the structure of N^t for all subsequent t is not affected by the tie-breaking rule. Hence, C^R must be well-defined in this case.

[15] The argument can easily be extended to a tie at any step t of the algorithm and to ties between any number of edges.

Case 2: There are two possible C^1-cycles - N_i^1 and N_k^1 depending on how the tie is broken. Suppose the tie is broken in favour of ij so that N_i^1 forms a supernode. Then, $\{k\}$ forms a singleton node in Stage 1. But, since $c^1_{kj} = 0$, $N_i^1 \cup N_k^1$ will form a supernode in step 2 of the algorithm. Notice that if the tie in step 1 was resolved in favour of kj, then again $N_i^1 \cup N_k^1$ would have formed a supernode in step 2 of the algorithm.

Also, the minimum cost incident edges of nodes outside N_i^1 and N_k^1 remain the same whether we break ties in favor of ij or kj. Hence, we get the same stages of the algorithm from stage 2 onwards. So, if $r \notin N_i^1 \cup N_k^1$, then for all $s \in N^0 \setminus \{r\}$, c^R_{sr} cannot be influenced by the tie-breaking rule.

Let $s \in N_i^1 \cup N_k^1$. Then, irrespective of how the tie is broken, $\delta^1_s = 0$, and hence c^R_{rs} cannot be influenced by the tie-breaking rule for any r.

Case 3: Suppose N_i^1 is a C^1-cycle containing i and j, but there is no C^1-cycle involving k. Since the minimum incident edge on N_i^1 corresponds to kj and $c^1_{kj} = 0$, $\delta^1_p = 0$ for all $p \in N_i^1$ irrespective of how the tie is broken. So, for all $s \in N_i^1$, c^R_{rs} is unaffected by the tie-breaking rule for all r.

These arguments prove that Equation 4 gives a well-defined irreducible cost matrix. ■

In the next two lemmas, we want to show (i) that the total cost of the minimum cost arborescences corresponding to cost matrices C and C^R is the same, and (ii) that the set $M(C^R)$ is "large" in the following sense - choose any ordered pair ij such that $j \neq 0$. Then, there is some MCA $g \in M(C^R)$ such that $ij \in g$. The proofs of these results involve a similar construction which we now describe.

In particular, we associate a cost matrix to *each* stage of the algorithm. Apart from being used in the proofs of the next two lemmas, these cost matrices also give an alternate interpretation of the irreducible cost matrix.

Let C be any cost matrix, and as usual, let C^R be the irreducible cost matrix corresponding to C.

Consider stage t of the recursive algorithm. The set of nodes in stage t is N^t. Consider $N_i^t, N_j^t \in N^t$ and let them be $\hat{t}$-neighbors, i.e., in some stage $\hat{t} \in \{t+1, \ldots, T-1\}$ nodes N_i^t and N_j^t become part of the same supernode for the first time. Say that they are T-neighbors if they are not $\hat{t}$ neighbors for some $\hat{t} \leq T-1$. Now, define

$$\widehat{c}_{N_i^t N_j^t} = \sum_{t'=t}^{\hat{t}-1} \delta^{t'}_{N_j^t}.$$

This defines a cost matrix $\widehat{C}^t$ on nodes N^t for stage t of the algorithm. Note that $C^R = \widehat{C}^0$. From the algorithm, the total cost of an MCA with nodes N corresponding to cost matrix C^t is equal to the total cost of an MCA with nodes N^t corresponding to cost matrix $\widetilde{C}^t$. Moreover, $\widehat{C}^t$ is the irreducible cost matrix of $\widetilde{C}^t$.

LEMMA **2** *Suppose C^R is the irreducible cost matrix corresponding to cost matrix C. Then, $T(C) = T(C^R)$.*

Proof: We prove the result by using induction on the number of stages of the algorithm. If $T = 1$, then $c^R_{ij} = \Delta^0_j$ for all edges ij. By definition of the irreducible cost matrix, $T(C) = \sum_{j \in N} \Delta^0_j = T(C^R)$.

Now, assume the lemma holds for any cost matrix that takes less than t stages where $t > 1$. We show that the lemma holds for any cost matrix C that takes t stages. Consider cost matrix $\widehat{C}^1$, which is the irreducible cost matrix of $\widetilde{C}^1$. Note that the algorithm takes $t-1$ stages for cost matrix $\widetilde{C}^1$, applied to nodes in N^1. Hence, by our induction hypothesis and using the fact $T(C^1) = T(\widetilde{C}^1)$, we get

$$T(C^1) = T(\widetilde{C}^1) = T(\widehat{C}^1). \tag{5}$$

Now, consider the cost matrix $\bar{C}$ defined as follows: $\bar{c}_{ij} = 0$ if $N^1_i = N^1_j$ and $\bar{c}_{ij} = \hat{c}^1_{N^1_i N^1_j}$ otherwise. Thus, $\bar{C}$ is a cost matrix on N^0 with edges in supernodes in stage 1 having zero cost and other edges having the same cost as in cost matrix $\widehat{C}^1$. Clearly, $T(\hat{C}^1) = T(\bar{C})$ (since edges inside a supernode in stage 1 have zero cost in $\bar{C}$). Using Equation 5,

$$T(C^1) = T(\bar{C}). \tag{6}$$

But cost matrices $\bar{C}$ and C^R differ as follows: for any edge ij in the original graph

$$\bar{c}_{ij} + \Delta^0_j = c^R_{ij}.$$

This implies that

$$T(C^R) = T(\bar{C}) + \sum_{j \in N} \Delta^0_j. \tag{7}$$

Similarly, C^1 and C differ as follows: for any edge ij

$$c^1_{ij} + \Delta^0_j = c_{ij}.$$

This implies that

$$T(C) = T(C^1) + \sum_{j \in N} \Delta^0_j. \tag{8}$$

Using Equations 6, 7, and 8,

$$T(C) = T(C^1) + \sum_{j \in N} \Delta^0_j = T(\bar{C}) + \sum_{j \in N} \Delta^0_j = T(C^R).$$

This completes the proof. ■

Lemma **3** *Choose any $j \in N$ and $i \in N^+$. Then, there is some $g \in M(C^R)$ such that $ij \in g$.*

Proof: We use induction on the number of stages T of the algorithm for cost matrix C. Suppose $T = 1$. By definition of the irreducible cost matrix, $c^R_{pq} = \Delta^0_q$ for every ordered pair pq where $p \in N^+$ and $q \in N$. Clearly, the total cost of an MCA corresponding to cost matrix C^R is $T(C^R) = \sum_{q \in N} \Delta^0_q$. Consider the arborescence which connects every agent except agent j to the source and agent j via agent i. The total cost of this arborescence corresponding to cost matrix C^R is $\sum_{q \in N} \Delta^0_q = T(C^R)$. Hence, this is an MCA corresponding to cost matrix C^R. Thus, every edge ij is used in some MCA corresponding to cost matrix C^R.

Now, suppose the claim holds for any cost matrix that takes less than t stages in the algorithm, where $t > 1$. We show that the claim holds for any cost matrix C that takes t stages in the algorithm. Consider cost matrix $\widehat{C}^1$, which is the irreducible cost matrix of $\widetilde{C}^1$. Since the algorithm takes $t-1$ stages for cost matrix $\widetilde{C}^1$ every edge $N^1_i N^1_j$ belongs to an MCA corresponding to $\widehat{C}^1$ from the induction hypothesis.

Now, choose any ordered pair ij, with $i \in N^+$ and $j \in N$. We consider two possible cases.

Case 1: Suppose i and j do not belong to the same supernode in N^1. Let i in N^1_i and $j \in N^1_j$. Let g^1 be the MCA corresponding to $\widehat{C}^1$ which contains the edge $N^1_i N^1_j$. Now, for each $p \in N^1_i, q \in N^1_j$,

$$c^R_{pq} = \widehat{c}_{N^1_i N^1_j} + \Delta^0_q \tag{9}$$

Also, for each $k, l \in N^1_j$,

$$c^R_{kl} = \Delta^0_l \tag{10}$$

Consider the total cost of connecting any $p \in N^1_i$ to some $q \in N^1_j$, and then connecting all nodes in N^1_j other than q to q. From equations 9 and 10, this equals

$$\begin{aligned} Q &= \widehat{c}_{N^1_i N^1_j} + \Delta^0_q + \sum_{r \in N^1_j, r \neq q} \Delta^0_r \\ &= \widehat{c}_{N^1_i N^1_j} + \sum_{r \in N^1_j} \Delta^0_r \end{aligned}$$

Hence, Q is independent of both the node $p \in N^1_i$ and the node $q \in N^1_j$; that is the point of exit from N^1_i and the point of entry into N^1_j. This establishes that there must be some $g \in M(C^R)$ which connects $i \in N^1_i$ to j in N^1_j.

Case 2: Suppose i, j belong to the same supernode N^1_k. From the previous paragraph, we know that there must be some $g \in M(C^R)$ which enters N^1_k at i and then connects j to i. ■

The next lemma shows that the additional cost (according to C^R) that any i imposes on a subset S not containing i is precisely the minimum amongst the costs of all incoming edges of i from S.

LEMMA 4 *Consider any S which is a proper subset of N and any $i \notin S$. Then,*

$$c^R(S \cup \{i\}) - c^R(S) = \min_{k \in S} c^R_{ki}$$

Proof: Suppose the recursive algorithm for N corresponding to C^R ends in one step. Then, for any $i \in N$ and all other $k \in N^0_k$, $c^R_{ki} = \Delta^0_i$. Clearly, the lemma must be true in this case.

Suppose the algorithm for N terminates in more than one step. Pick any S and $i \notin S$. Let $T \equiv S \cup \{i\}$, and $T^1 = \{T^1_1, \ldots, T^1_K\}$, where each $T^1_k = N^1_k \cap T$, N^1_k being a supernode in N^1. Without losss of generality, let

$$\min_{k \in S} c^R_{ki} = c^R_{k^*i}$$

Suppose $k^* \in N^1_k$ and $i \in N^1_i$ with $N^1_i \setminus \{i\} \neq \emptyset$ and $N^1_i \neq N^1_k$. But for any $j \in N^1_i \setminus \{i\}$, we have $c^R_{ji} \leq c^R_{k^*i}$. In that case, we set $k^* = j$. Hence, there are two possibilities without loss of generality. Either k^* and i belong to the some T^1_k or $\{i\} \in T^1$.

CASE 1 : k^* and i belong to some T^1_k. Note that in this case, c^R_{ki} is the same for all k in $T^1_k, k \neq i$. From the proof of Lemma 3, there is an MCA g for T and C^R such that T^1_k is the last supernode to be entered. Furthermore, g can enter T^1_k at some node k distinct from i, and then connect all other nodes in T^1_k to k. So, i is a *leaf* of g since $|\delta^-(j,g)| = 1$. Hence, the lemma is true in this case.

CASE 2 : i forms a singleton node in T^1. Then, $c^R_{k^*i}$ is the cost of connecting $\{i\}$ to *every* other supernode in T^1. Consider the MCA g for T^1 and C^R which enters $\{i\}$ last. Then, the lemma is true in this case too. ■

We use this lemma to show that the cost game corresponding to C^R is concave.

LEMMA 5 *The cost game corresponding to C^R is concave.*

Proof: Consider any S, T such that $S \subset T \subseteq N$ with $i \notin T$. We show that

$$c^R(S \cup \{i\}) - c^R(S) \geq c^R(T \cup \{i\}) - c^R(T)$$

By Lemma 4,

$$c^R(S \cup \{i\}) - c^R(S) = \min_{k \in S} c^R_{ki}$$

and

$$c^R(T \cup \{i\}) - c^R(T) = \min_{k \in T} c^R_{ki}$$

Since $S \subset T$, the lemma follows. ■

We now show that a "small" change in C produces a small change in C^R. In other words, C^R changes continuously with C.

LEMMA **6** *Suppose C and $\bar{C}$ are such that $c_{kl} = \bar{c}_{kl}$ for all $kl \neq ij$ and $\bar{c}_{ij} = c_{ij} + \epsilon$ for some $\epsilon > 0$. If C^R and $\bar{C}^R$ are the irreducible matrices corresponding to C and $\bar{C}$, then*

(i) $0 \leq \bar{c}^R_{kj} - c^R_{kj} \leq \epsilon$ for all $k \neq j$ and $-\epsilon \leq \bar{c}^R_{kl} - c^R_{kl} \leq \epsilon$ for all $k \neq l \neq j$,

(ii) for all $l \in N \setminus \{j\}$, for all $S \subseteq N^+ \setminus \{l, j\}$,

$$\min_{k \in S} \bar{c}^R_{kj} - \min_{k \in S} c^R_{kj} \geq \min_{k \in S} \bar{c}^R_{kl} - \min_{k \in S} c^R_{kl} \tag{11}$$

Proof: Say that two cost matrices C and $\bar{C}$ are *stage equivalent* if ties can be broken in the recursive algorithm such that the number of stages and partitions of nodes in each stage are the same for C and $\bar{C}$.

PROOF OF (i): We first prove this for the case when C and $\bar{C}$ are stage equivalent. Consider node j such that $\bar{c}_{ij} - c_{ij} = \epsilon > 0$. Consider any $k \neq j$. Since C and $\bar{C}$ are stage equivalent, if k and j are t-neighbors in cost matrix C, then they are t-neighbors in cost matrix $\bar{C}$. We consider two possible cases.

CASE 1: Edge ij is the minimum cost incident edge of some supernode containing j in some stage t of the algorithm for cost matrix C, and hence for cost matrix $\bar{C}$ since they are stage equivalent. Then δ^t_j increases by ϵ. But δ^{t+1}_j (if stage $t+1$ exists) decreases by ϵ. Hence, the irreducible cost of no edge can increase by more than ϵ and the irreducible cost of no edge can decrease by more than ϵ. Moreover, we prove that for any edge kj, the irreducible cost cannot decrease. To see this, note that the irreducible cost remains the same if kj are t' neighbors and $t' \leq t$ or $t' > t+1$. If $t' = t+1$, then the irreducible cost of kj only increases.

CASE 2: Edge ij is not the minimum cost incident edge of any supernode containing j in any stage of the algorithm for cost matrix C, and hence for cost matrix $\bar{C}$. In that case, δ^t_j remains the same for all t. Hence $C^R = \bar{C}^R$.

Examining both the cases, we conclude that $0 \leq \bar{c}^R_{kj} - c^R_{kj} \leq \epsilon$ for all $k \neq j$. Also, $-\epsilon \leq (\bar{c}^R_{kl} - c^R_{kl}) \leq \epsilon$ for all $k \neq l \neq j$.

We complete the proof by arguing that cost of edge ij can be increased from c_{ij} to $\bar{c}_{ij}$ by a finite sequence of increases such that cost matrices generated in two consecutive sequences are stage equivalent.

Define *ranking* of an edge ij in cost matrix C as $rank^C(ij) = |\{kl : c_{kl} > c_{ij}\}|$. Clearly, two edges ij and kl have the same ranking if and only if $c_{ij} = c_{kl}$. Note that if rankings of edges do not change from C to $\bar{C}$, then we can always break ties in the same manner in the recursive algorithm in C and $\bar{C}$, and thus C and $\bar{C}$ are stage equivalent.

Suppose $rank^C(ij) = r > rank^{\bar{C}}(ij) = \bar{r}$. Consider the case when $\bar{r} = r - 1$. This means that a unique edge $kl \neq ij$ exists such that $c_{kl} > c_{ij}$ but $\bar{c}_{kl} = c_{kl} \leq \bar{c}_{ij}$. Consider an *intermediate* cost matrix $\hat{C}$ such that $\hat{c}_{ij} = c_{kl} = \bar{c}_{kl}$ and $\hat{c}_{pq} = c_{pq} = \bar{c}_{pq}$ for all edges $pq \neq ij$. In the cost matrix $\hat{C}$, one can break ties in the algorithm such that one chooses ij over kl everywhere. This will generate the same stages of the algorithm with same partitions of nodes in every stage for cost matrix C and $\hat{C}$. Hence, C and $\hat{C}$ are stage equivalent. But we can also break the ties in favor of edge kl everywhere, and this will generate the same set of stages and partitions as in cost matrix $\bar{C}$. This shows that $\hat{C}$ and $\bar{C}$ are also stage equivalent.[16].

If $\bar{r} < r - 1$, then we increase the cost of edge ij from c_{ij} in a finite number of steps such that at each step, the ranking of ij falls by exactly one.

Proof of (ii): For simplicity, we only consider the case where C and $\bar{C}$ are stage-equivalent. As argued in the proof of (i), the argument extends easily to the case when they are not stage-equivalent.

Let ij be the minimum cost incident edge of N_k^t. Then, for any $p \in N_k^t$,

$$\bar{c}_{qp}^R > c_{qp}^R \text{ implies } p, q \text{ are } (t+1)\text{-neighbours} \tag{12}$$

Also, for all $p \in N_k^t$ and all q which are $(t+1)$-neighbours,

$$\bar{c}_{qp}^R = c_{qp}^R + \epsilon$$

So, to prove that equation 11 is true, we only need to consider any $l \in N_k^t, l \neq j$. Consider any $S \subseteq N^+ \setminus \{l, j\}$, and suppose $\min_{k \in S} \bar{c}_{kl}^R > \min_{k \in S} c_{kl}^R$. Then, from Equation 12, S must contain some q which is a $(t+1)$-neighbour of l. Moreover, since the irreducible cost of edges which are t'-neighbours are higher than those which are $(t'-1)$-neighbours for all t', S cannot contain any neighbours which are t'-neighbours of l for $t' \leq t$. This is because if S did contain some m which was a t'-neighbour with $t' \leq t$, then $\min_{k \in S} \bar{c}_{kl}^R$ would not be attained at a $(t+1)$-neigbour of l. But, then equation 12 establishes that

$$\min_{k \in S} \bar{c}_{kl}^R - \min_{k \in S} c_{kl}^R = \min_{k \in S} \bar{c}_{kj}^R - \min_{k \in S} c_{kj}^R = \epsilon$$

[16] Note that this does not imply C and $\bar{C}$ are stage equivalent.

This completes the proof of the lemma. ■

Remark 2 *So, a small change of ϵ in the cost of some edge ij cannot change the irreducible cost of any edge by more than ϵ. Hence, C^R changes continuously with C.*

We identify a cost allocation rule f^* with the Shapley value of the cost game (N, C^R). So,

$$f^*(C) \equiv Sh(N, C^R)$$

The main result of the paper is to show that f^* satisfies CS, DSCM, and Con.

Theorem 4 *The cost allocation f^* satisfies CS, DSCM, and Con.*

Proof: Take any cost matrix C whose irreducible cost matrix is C^R. From Lemma 5, the game (N, C^R) is concave. Hence, the Shapley value of (N, C^R) is in the core of the game (N, C^R). Of course, $c_{ij}^R \leq c_{ij}$ for all edges ij. Using Lemma 2 it follows that the core of the game (N, C^R) is contained in the core of the game (N, C). Hence, f^* satisfies CS.

By Lemma 4, $c^R(S \cup \{k\}) - c^R(S) = \min_{j \in S} c_{jk}^R$ for any $S \subseteq N$ with $k \notin S$. By Lemma 6, C^R changes continuously with C. Hence, the cost game (N, c^R) also changes continuously with C. The continuity of the Shapley value with respect to the cost game establishes that f^* satisfies continuity.

We use Lemma 4 and Equation 11 to prove that f^* satisfies DSCM. Consider two cost matrices C and $\bar{C}$ as in Lemma 6. From Lemma 4, the marginal costs of any l to a coalition S for C^R and $\bar{C}^R$ are given by

$$\min_{k \in S} c_{kl}^R \text{ and } \min_{k \in S} \bar{c}_{kl}^R$$

Using the formula for the Shapley value and Equation 11, it is straightforward to verify that DSCM is satisfied. ■

In the minimum cost spanning tree problem, the folk solution satisfies a *Solidarity* axiom which requires the following. If the cost of an edge goes up, then the cost share of *every* node should increase or remain the same. This is, of course, stronger than the cost monotonicity axiom of Dutta and Kar (2004), and was put forward by Bergantinos and Vidal-Puga (2007a). The following example shows that f^* does not satisfy the directed version of the Solidarity axiom.

Example 1 *Let $N = \{1, 2\}$, and consider cost matrices C and $\bar{C}$ as follows*

$$c_{12} = c_{21} = 1, c_{01} = 2, c_{02} = 3, \bar{c}_{21} = 1.5, \bar{c}_{ij} = c_{ij}, \textit{for all edges } ij \neq 21$$

The corresponding irreducible cost matrices are

$$c_{12}^R = c_{21}^R = 1, c_{01}^R = c_{02}^R = 2, \bar{c}_{21}^R = 1.5, \bar{c}_{12}^R = 1, \bar{c}_{01}^R = 2, \bar{c}_{02}^R = 1.5$$

Then, $f^*(C) = (1.5, 1.5)$ and $f^*(\bar{C}) = (1.75, 1.25)$.

We now show that f^* satisfies Ranking.

THEOREM **5** *The allocation rule f^* satisfies ETE and Ranking.*

Proof: Since the Shapley value is symmetric, f^* satisfies ETE.

Consider a cost matrix C such that for $i, j \in N$, we have $c_{ik} = c_{jk}$ and $c_{ki} > c_{kj}$ for all $k \neq i, j$ and $c_{ji} > c_{ij}$. Consider another cost matrix $\hat{C}$ such that $\hat{c}_{ik} = \hat{c}_{jk} = c_{ik} = c_{jk}$ and $\hat{c}_{ki} = \hat{c}_{kj} = c_{kj} < c_{ki}$ for all $k \neq i, j$ and $\hat{c}_{ij} = \hat{c}_{ji} = c_{ij} < c_{ji}$. By ETE, $f_j(\hat{C}) = f_i(\hat{C})$.

Now, let $\epsilon = \min_{k \neq i}[c_{ki} - \hat{c}_{ki}]$. Note that $\epsilon > 0$ by assumption. Consider a cost matrix $\bar{C}$ defined as follows: $\bar{c}_{ki} = \hat{c}_{ki} + \epsilon$ for all $k \neq i$ and $\bar{c}_{pq} = \hat{c}_{pq}$ for all p, q with $q \neq i$. So, we increase cost of incident edges on i from $\hat{C}$ to $\bar{C}$ by the same amount ϵ, whereas costs of other edges remain the same.

Let $\hat{c}_{ki} = \min_{p \neq i} \hat{c}_{pi}$. By construction, $\bar{c}_{ki} = \min_{p \neq i} \bar{c}_{pi} = \hat{c}_{ki} + \epsilon$. Also, $\bar{C}^1 = \hat{C}^1$. Hence, $\bar{c}^R_{pi} = \hat{c}^R_{pi} + \epsilon$ for all $p \neq i$. Thus, $f^*_i(\bar{C}) = f^*_i(\hat{C}) + \epsilon$. But the cost of the MCA has increased by ϵ from $\hat{C}$ to $\bar{C}$. So, $f^*_k(\bar{C}) = f^*_k(\hat{C})$ for all $k \neq i$. Hence, $f^*_i(\bar{C}) > f^*_i(\hat{C}) = f^*_j(\hat{C}) = f^*_j(\bar{C})$. Since f^* satisfies DSCM, $f^*_i(C) - f^*_j(C) \geq f^*_i(\bar{C}) - f^*_j(\bar{C}) > 0$. This implies that $f^*_i(C) > f^*_j(C)$. ■

Notice that the condition of Ranking requires that if the incoming edges of node i cost strictly more than the corresponding incoming edges for j while corresponding outgoing edges cost the same, then the cost allocated to i should be strictly higher than the cost allocated to j. But, now suppose *both* incoming *and* outgoing edges of i cost strictly more than those of j. Perhaps, one can argue that if the outgoing edges of i cost more than the outgoing edges of j, then i is less "valuable" in the sense that i is going to be used less often in order to connect to other nodes. Hence, in this case too, i should pay strictly more than j. However, it turns out that f^* does not satisfy this modified version of Ranking.[17] This is demonstrated below.

EXAMPLE **2** *Let $N = \{1, 2, 3\}$. Figure 5 shows a cost matrix C (assume $e > 0$ in Figure 5) and its associated irreducible cost matrix, the latter being shown on the right. It is easy to see that $f^*_1(C) = e, f^*_2(C) = f^*_3(C) = 1/2$. So, for $e < 1/2$, agent 1 pays less than agent 2, though agent 2 has strictly lower incoming and outgoing edge costs than agent 1.*

We do not know whether there are other rules which satisfy the basic axioms and this modified version of Ranking.

[17] We do not define this axiom formally.

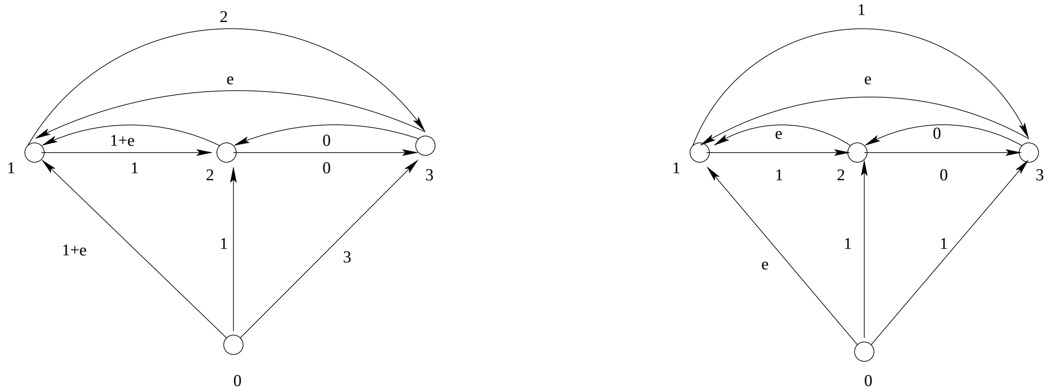

Figure 5: Example illustrating violation of the modified version of Ranking

References

A. Bogomolnaia, R. H., , and H. Moulin (2008): "Sharing the Cost of a Capacity Network," Working Paper, Rice University.

Bergantinos, G. and A. Kar (2008): "Obligation Rules," Working Paper, Universidade Vigo.

Bergantinos, G. and J. J. Vidal-Puga (2007a): "A Fair Rule for Minimum Cost Spanning Tree Problems," *Journal of Economic Theory*, 137, 326–352.

——— (2007b): "The Optimistic TU Game in Minimum Cost Spanning Tree Problems," *International Journal of Game Theory*, 36, 223–239.

——— (2007c): "Several Approaches to the Same Rule in Minimum Cost Spanning Tree Problems," Working Paper, Universidade Vigo.

Bird, C. G. (1976): "On Cost Allocation of a Spanning Tree: A Game Theoretic Approach," *Networks*, 6, 335–350.

Bogomolnaia, A. and H. Moulin (2008): "Beyond the Folk Solution in the Minimum Cost Spanning Tree Problem," Working Paper, Rice University.

Branzei, R., S. Moretti, H. Norde, and S. Tijs (2004): "The P-Value for Cost Sharing in Minimum Cost Spanning Tree Situations," *Theory and Decision*, 56, 47–61.

——— (2005): "The Bird Core for Minimum Cost Spanning Tree Problems Revisited: Monotonicity and Additivity Aspects," Working Paper, Tilburg University (CentER DP).

Chu, Y. J. and T. H. Liu (1965): "On the Shortest Arborescence of a Directed Graph," *Science Sinica*, 14, 1396–1400.

Dutta, B. and A. Kar (2004): "Cost Monotonicity, Consistency and Minimum Cost Spanning Tree Games," *Games and Economic Behavior*, 48, 223–248.

Edmonds, J. (1967): "Optimum Branchings," *Journal of Reseseach National Bureau Standards*, 71B, 233–240.

Feltkamp, V., S. Muto, and S. Tijs (1994): "On the Irreducible Core and the Equal Remaining Obligations Rule of Minimum Cost Spanning Extension Problems," Working Paper, Tilburg University (CentER DP).

Herzog, S., S.Shenker, and D.Estrin (1997): "Sharing the Cost of Multicast Trees: An Axiomatic Analysis," *IEEE/ACM Transactions on Networking*, 847–860.

Kar, A. (2002): "Axiomatization of the Shapley Value on Minimum Cost Spanning Tree Games," *Games and Economic Behavior*, 38, 265–277.

Norde, H., S. Morettie, and S. Tijs (2001): "Minimum Cost Spanning Tree Games and Population Monotonic Allocation Schemes," *European Journal of Operational Research*, 154, 84–97.

Shapley, L. (1971): "Cores of Convex Games," *International Journal of Game Theory*, 1, 11–26.